

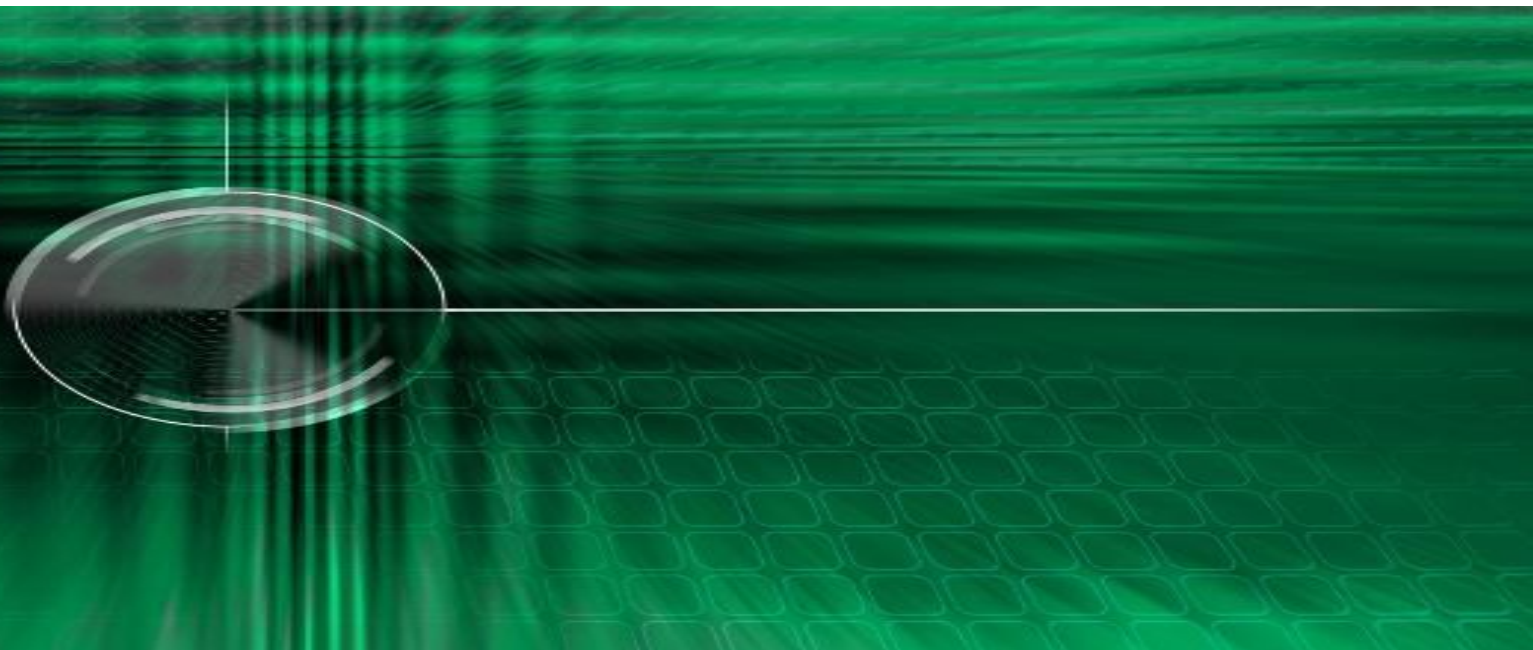


Software Engineering
Competence Center

TUTORIAL

An Introduction to Cloud Computing Concepts

Practical Steps for Using Amazon EC2 IaaS Technology



Ahmed Mohamed Gamaleldin

Senior R&D Engineer-SECC

ahmed.gamal.eldin@itida.gov.eg

Abstract

Cloud computing is the evolution of a variety of technologies that have come together to change an organization's approach for building an IT infrastructure. There is nothing fundamentally new in any of the technologies that make up cloud computing as most of these technologies have been used for ages. The cloud computing term describes a variety of different types of computing concepts that involve a large number of computers connected through a real-time communication network (typically the Internet).

Cloud computing relies on sharing of various resources (e.g., networks, servers, storage, applications, and services) to achieve coherence and economies of scale, and gives the highest interest to how to maximize the effectiveness of utilization of the shared resources.

This tutorial provides an introduction to the cloud computing concepts and terminologies. In addition, it'll give a step by step example for creating a cloud instance using Amazon EC2 Infrastructure as a service (IaaS) technology.

Keywords: Cloud, Cloud Computing, Amazon EC2, Infrastructure as a Service (IaaS), Web Service.

Table of Contents

1. Introduction	4
2. Essential Characteristics of the Cloud	5
3. Cloud Services Models	5
4. Infrastructure as a Service (IaaS)	7
5. Amazon Web Services (AWS)	8
5.1. Amazon Elastic Cloud Computing (EC2)	8
5.1.1. Amazon EC2 Concepts	9
5.1.2. Amazon EC2 Access	10
6. Amazon EC2 step by step	10
7. Summary	18
8. References	19
9. Abbreviations	19

1. Introduction

Cloud computing can be defined as a model for enabling ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort from the user side and minimal service provider interaction.

Cloud computing is considered the evolution of a variety of technologies that have come together to change an organizations' approach for building their IT infrastructure. Actually, there is nothing new in any of the technologies that are used in the cloud computing where most of these technologies have been known for ages. It is all about making them all accessible to the masses under the name of cloud computing. Cloud is not simply the latest term for the Internet, though the Internet is a necessary foundation for the cloud, the cloud is something more than the Internet. The cloud is where you go to use technology when you need it, for as long as you need it. You do not install anything on your desktop, and you do not pay for the technology when you are not using it.

The cloud can be both software and infrastructure. It can be an application you access through the Web or a server like Gmail and it can be also an IT infrastructure that can be used as per user's request. Whether a service is software or hardware, the following is a simple test to determine whether that service is a **cloud service**:

If you can walk into any place and sit down at any computer without preference for operating system or browser and access a service, that service is cloud-based. Generally, there are three measures used to decide whether a particular service is a cloud service or not:

- The service is accessible via a web browser or web services API.
- Zero capital expenditure is necessary to get started.
- You pay only for what you use.

2. Essential Characteristics of the Cloud

We can summarize the essential characteristics of the Cloud as below:

- **On-demand self-service:** A service consumer can automatically make use of the computing capabilities, such as server processing time and network storage without requiring human interaction with each service's provider.
- **Broad network access:** Cloud capabilities (HW and SW) are available over the network and accessed through various platforms (e.g., mobile phones, laptops, and tablets).
- **Resource pooling:** The provider's computing resources (HW and SW) are pooled to serve multiple consumers using a *multi-tenant* model, with different physical and virtual resources dynamically assigned and reassigned according to users' demand. Multi-tenancy is the most important feature of the cloud-based application. It is characterized by the location independence feature in which the customer has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
- **Rapid elasticity:** Capabilities can be rapidly and elastically provisioned; it can be quickly scaled out, and quickly scaled in. For the user, the capabilities available for provisioning appear to be unlimited and can be purchased in any quantity at any time.
- **Measured Service:** Cloud systems automatically control and optimize resources use by leveraging a metering capability in which resources' usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service. The advantage here is that you are paying for exactly what you are using.

3. Cloud Services Models

The cloud services can be categorized into *software services* and *infrastructure or hardware services*. In terms of maturity, software in the cloud is much more evolved than hardware in the cloud.

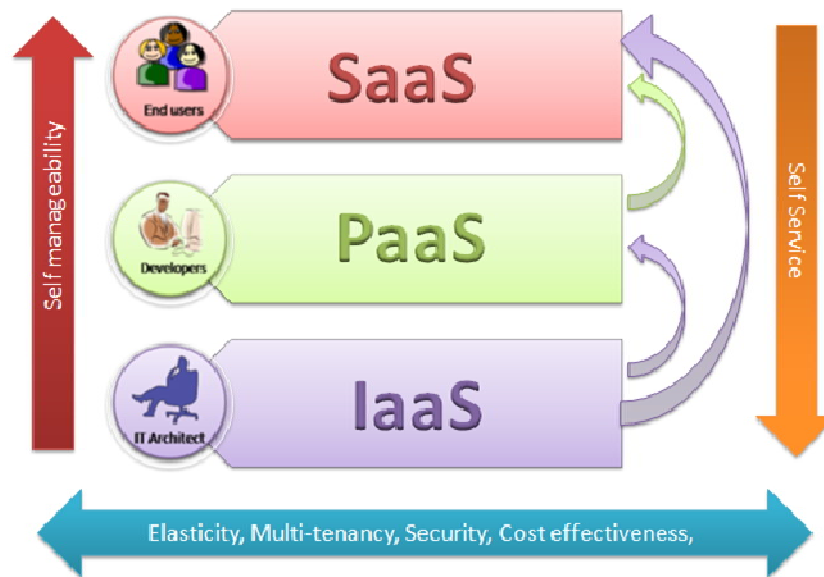


Figure 1. Cloud service models

Cloud Software as a Service (SaaS) is basically a term that refers to software in the cloud. It represents the capability provided to the consumer to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through an interface such as a web browser (e.g. web-based email like Gmail is a form of SaaS provided by Google). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities.

SaaS systems have some defining characteristics:

- **Availability via web browser**
SaaS software never requires the installation of software on your laptop or desktop. You access it through a web browser using open standards or a browser plug-in.
- **On-demand availability**
You should not have to go through a sales process to gain access to SaaS-based software. Once you have access, you should be able to go back into the software any time, from anywhere.
- **Payment terms based on usage**
SaaS does not need any infrastructure investment or complex setup, so you should not have to pay any massive setup fees. You should simply pay for the parts of the service you use as you use them. When you no longer need those services, you simply stop paying.
- **Minimal IT demands**
SaaS systems don't require a high technical knowledge for their configuration.

Cloud Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure, but has control over the deployed applications and possibly application hosting environment configurations.

Cloud Infrastructure as a Service (IaaS). The capability provided to the consumer is to make use of processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications.

4. Infrastructure as a Service (IaaS)

The focus of this tutorial is on the IaaS service model. Amazon Web Services (AWS) is one of the major players in this area. The AWS is based on pure virtualization, it owns all the hardware and controls the network infrastructure and you own everything from the guest operating system up. You request virtual instances on-demand and let them go when you are done.

AppNexus [3] represents a different approach to this problem. As with AWS, AppNexus enables you to gain access to servers on demand. However, it provides dedicated servers with virtualization on top. You have the confidence in knowing that your applications are not fighting with anyone else for resources and that you can meet any requirements that demand full control over all physical server resources.

Hybrid computing takes advantage of both approaches, offering virtualization when appropriate and a dedicated hardware when appropriate. In addition, most hybrid vendors such as Rackspace [4] and GoGrid [5] base their model on the idea that users still want a traditional data center and dedicated storage, but they just want it in the cloud.

This tutorial focuses on the Amazon AWS and provides a practical example about using the Amazon EC2 IaaS solution.

5. Amazon Web Services (AWS)

AWS is Amazon's umbrella description of all of their web-based technology services. It encompasses a wide variety of services, all of which fall into the concept of cloud computing like:

- Amazon Elastic Cloud Compute (Amazon EC2)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Queue Service (Amazon SQS)
- Amazon CloudFront
- Amazon SimpleDB

5.1. Amazon Elastic Cloud Computing (EC2)

Amazon EC2[2] is the heart of the Amazon cloud. It provides a web services API for provisioning, managing, and de-provisioning of virtual servers inside the Amazon cloud. In other words, any application anywhere on the Internet can launch a virtual server in the Amazon cloud with a single web services call.

At the time of this tutorial, Amazon's EC2 U.S. footprint has three data centers on the east coast of the U.S. and two in western Europe[1]. You can sign up separately for an Amazon European data center account, but you cannot mix U.S. and European environments. The servers in these environments run a highly customized version of tools that are using the virtualization concept and enable the dynamic provisioning and de-provisioning of servers. When you want to start up a virtual server in the Amazon environment, you launch a new instance based on a predefined *Amazon machine image* (AMI). The AMI includes your operating system and any other prebuilt software. Most people start with a standard AMI based on their favorite operating system, customize it, create a new image based on their preferences, and then launch their servers based on their custom images.

Many competitors to Amazon also provide persistent internal storage for nodes to make them operate more like a traditional data center.

In addition, servers in EC2 like any other server on the Internet can access Amazon S3 for cloud-based persistent storage. EC2 servers in particular achieve cost savings and greater efficiencies in accessing S3 [1].

When most people think of the Amazon cloud, they are thinking about Amazon EC2. EC2 represents your virtual network with all of the virtual

servers running inside that network. When you use EC2, you will be able to use S3 to store your machine images and also for other storage needs.

5.1.1. Amazon EC2 Concepts

Amazon EC2 [2] is a bit more complex than S3. Figure 2 shows all of the concepts that make up Amazon EC2 and how they relate to each other.

The main concepts are:

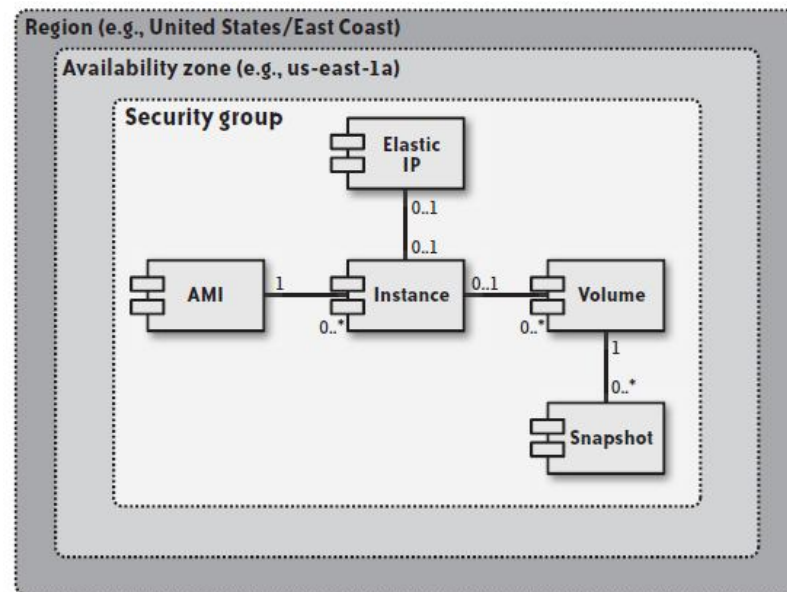


Figure 3. An overview of the components that support Amazon EC2

- **Instance**
An Amazon EC2 instance is a virtual server running your choice of guest operating system based on the machine image from which the instance was created.
- **Amazon Machine Image (AMI)**
A copy of your server that you can use to launch any number of instances. If you are familiar with the concept of ghosting, the machine image represents your ghost image from which you can build any number of servers. Minimally, a machine image will have the core operating system plus common preinstalled tools. Amazon has some prebuilt AMIs to get you started easily.
- **Elastic IP address**
This is simply a static IP address that is assigned to you. (The term "elastic" doesn't mean a dynamic address, it is a static one) By default, each Amazon instance comes with a dynamically assigned IP address that

may be reassigned to another user when your instance terminates. Elastic IP addresses are reserved to you and thus useful for instances that must always be accessible by the same static IP address.

- **Region**

A group of availability zones that form a single geographic cluster. Until now, Amazon's service level agreement (SLA) for EC2 guarantees 99.95% availability of at least two availability zones within a region over the course of a 12-month period [1].

- **Availability zone**

Almost analogous to a data center. Amazon currently has three zones in the U.S., all on the east coast. It also has two zones in western Europe. You may optionally define the availability zone into which you launch your instances to create a level of location redundancy for your applications.

- **Security group**

Very roughly analogous to a network segment protected by a firewall. You launch your instances into security groups and, in turn, the security groups define what can talk to your new instances and what is prohibited.

- **Block storage volume**

Like a SAN (storage area network), it provides block-level storage that you can mount from your EC2 instances. You can then format the volume as you want, or write raw data to the volume.

- **Snapshot**

You may take "snapshots" of your block volumes for backup or replication purposes. These snapshots are stored in Amazon S3, where they can be used to create new volumes.

5.1.2. Amazon EC2 Access

Like Amazon S3, the primary means of accessing Amazon EC2 is through a web services API. Amazon provides a number of interactive tools on top of their web services API, including:

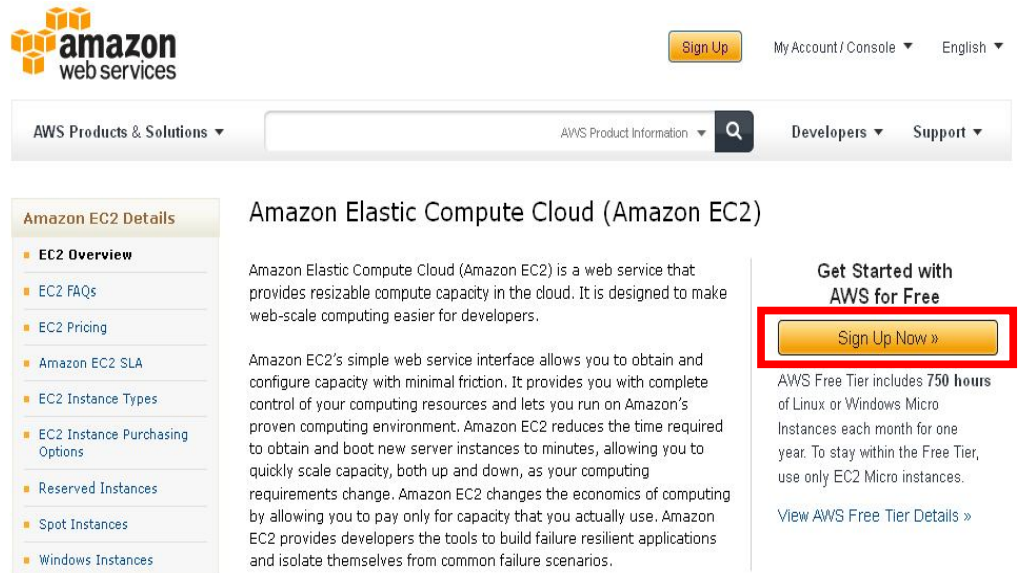
- The Amazon Web Services Management Console
- The ElasticFox Firefox plug-in
- The Amazon Command Line tools

The example explained here focuses on using the management console which is one of the easiest and user friendly means.

6. Amazon EC2 step by step

In this section, you will learn how to create and use the Amazon EC2 instance in a step by step.

1. Open the Amazon EC2 web page <http://aws.amazon.com/ec2/> and click the Sign up button.



2. Register with you e-mail.



3. Write down the data required as shown below.



Login Credentials

Use the form below to create login credentials that can be used for AWS as well as Amazon.com.

My name is:

My e-mail address is:

Type it again:

note: this is the e-mail address that we will use to contact you about your account

Enter a new password:

Type it again:

Continue

4. Add your payment information taking into account that it will not be charged as we are using a free service, but it is mandatory to enter your payment information.



Amazon Web Services Sign Up

CREATE ACCOUNT PAYMENT METHOD IDENTITY VERIFICATION CONFIRMATION

Your AWS account credentials have been created, but in order to begin using any of the services, you will need to provide your payment information and continue. There is no fee to sign up and you only pay for what you use.

Enter Your Payment Information Below

Your credit card will not be charged until you begin using AWS, and many of your applications and uses of AWS may be able to operate within the AWS free usage tier. If your monthly usage goes beyond the free tier, your AWS service charges will be billed to the credit card you provide below. [View detailed service pricing](#)

* required fields

Credit Card*:

Card Number*:

Cardholder's Name*:

Expiration Date*:

5. You'll be informed with a PIN number then; you'll get a phone call asking you to enter the PIN number to be identified correctly.
6. Click on launch Instance button.

EC2 Dashboard

Events

INSTANCES

Instances

Spot Requests

Reserved Instances

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Load Balancers

Key Pairs

Network Interfaces

Getting Started

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the region.

Service Health

Service Status

Current Status	Details
<div></div> Amazon EC2 (US East - N. Virginia)	Service is operating normally View complete service health details

Availability Zone Status

Current Status	Details
<div></div> us-east-1a	Availability zone is operating normally
<div></div> us-east-1b	Availability zone is operating normally
<div></div> us-east-1c	Availability zone is operating normally

- Enter your new instance name and select one of the available configuration (in this tutorial we are using the free Ubuntu 12.04.1 LTS configuration)

Create a New Instance

Select an option below:

Classic Wizard

Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.

Quick Launch Wizard

Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.

AWS Marketplace

AWS Marketplace is an online store where you can find and buy software that runs on AWS. Launch with 1-Click and pay by the hour.

Name Your Instance

SECC_R&D

Choose a Key Pair:

Select Existing

Create New

None

Name: e.g. myhosts

Download

Choose a Launch Configuration:

SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS

64 bit 32 bit

Ubuntu Server 12.04.1 LTS

64 bit 32 bit

Ubuntu Server 11.10

64 bit 32 bit

Microsoft Windows Server 2008 Base

64 bit 32 bit


Microsoft Windows Server 2008 R2 Base

64 bit 32 bit

Microsoft Windows Server 2008 R2 with SQL Server Express and IIS

64 bit 32 bit

- Enter the name of the .pem file, which is the identifier you'll need when connecting to the instance. After that click download button and

 An Introduction to Cloud Computing Concepts | Tutorial

Page 13

save this .pem file in a secure path on your hard disk. When you finish click on the continue button

Create a New Instance Cancel

Select an option below:

- Classic Wizard**
Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.
- Quick Launch Wizard**
Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.
- AWS Marketplace**
AWS Marketplace is an online store where you can find and buy software that runs on AWS. Launch with 1-Click and pay by the hour.

Name Your Instance: SECC_R&D Pick a meaningful name, e.g. Web Server

Choose a Key Pair:
Public/private key pairs allow you to securely connect to your instance after it launches.
☐ Select Existing ☒ Create New ☐ None

Name: SECC
Please note that you need to download the key pair before you can continue. Download

Choose a Launch Configuration:

Image	Operating System	Architecture	Free tier eligible
SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS boot with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, and Ruby 1.8.7 available	SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS boot with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, and Ruby 1.8.7 available	64 bit 32 bit	
Ubuntu Server 12.04.1 LTS	Ubuntu Server 12.04.1 LTS with support available from Canonical (http://www.ubuntu.com/cloud/services).	64 bit 32 bit	Free tier eligible
Ubuntu Server 11.10	Ubuntu Server version 11.10, with support available from Canonical (http://www.ubuntu.com/cloud/services).	64 bit 32 bit	Free tier eligible
Microsoft Windows Server 2008 Base	Microsoft Windows 2008 R1 SP2 Datacenter edition. [English]	64 bit 32 bit	Free tier eligible
Microsoft Windows Server 2008 R2 Base	Microsoft Windows 2008 R2 SP1 Datacenter edition and 64-bit architecture. [English]	64 bit 32 bit	Free tier eligible
Microsoft Windows Server 2008 R2 with SQL Server Express and IIS	Microsoft Windows 2008 R2 SP1 Datacenter edition and 64-bit architecture. [English]	64 bit 32 bit	Free tier eligible

Note: You can customize your settings in the next step. Continue

[Submit Feedback](#) [Getting Started Guide](#)

9. Click on the launch button

Create a New Instance Cancel

Ubuntu Server 12.04.1 LTS (ami-3d4ff254)
Platform: Ubuntu Architecture: x86_64
Ubuntu Server 12.04.1 LTS with support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Please review your settings and click **Launch** to finish or **Edit details** to make changes.

Instance Details

Property	Value
Name	SECC_R&D
Type	t1.micro
Detailed Monitoring	No
Availability Zone	No preference
Shutdown Behaviour	Stop
Termination Protection	No
Launch into a VPC	No

Security Details

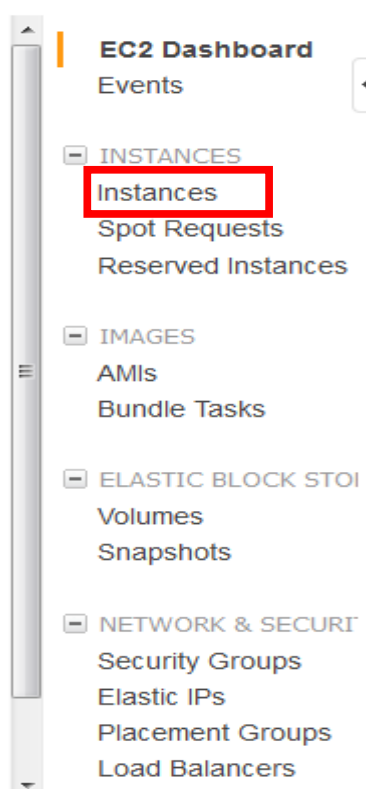
Property	Value
Key Pair	SECC
Security Group	quicklaunch-1

Advanced Details

Property	Value
Kernel ID	Default
Ramdisk ID	Default
User Data	
IAM Role	

[Go Back](#) [Edit details](#) Launch

10. Click on Instances from the left pane.



You'll find your created instance as shown below and it is running by default

The screenshot shows the AWS Management Console's EC2 Instances page. At the top, there are buttons for 'Launch Instance' and 'Actions'. Below these are filters for 'Viewing: All Instances' and 'All Instance Types', along with a search bar. The page shows '1 to 1 of 1 Instances'. The instance list table is as follows:

	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status	Monitoring	
<input type="checkbox"/>	SECC_R&D	i-0465447a	ami-3d4ff254	ebs	t1.micro	running	initializing...	none	basic	

11. Configure your security options by clicking the security groups from the left pane and check the "quicklaunch-1" security configuration (Created by default).

1. Create Security Group

2. Delete

Viewing: EC2 Security Groups

Name	VPC ID	Description
default		default group
<input checked="" type="checkbox"/> quicklaunch-1		quicklaunch-1

1 Security Group selected

Security Group: quicklaunch-1

Details Inbound

Create a new rule: Custom TCP rule

Port range: (e.g., 80 or 49152-65535)

Source: 0.0.0.0/0 (e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

+

TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete

+

12. Add the ports numbers of the most common applications in the port range entry one by one like:

- Port 80 and 8080 for HTML applications
- Port 21 for FTP
- Port 3306 for MySQL server application

Details Inbound*

Create a new rule: Custom TCP rule

Port range: (e.g., 80 or 49152-65535)

Source: 0.0.0.0/0 (e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

+

TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete
8080 (HTTP*)	0.0.0.0/0	Delete
3306 (MYSQL)	0.0.0.0/0	Delete
443 (HTTPS)	0.0.0.0/0	Delete

+

13. Select your instance by checking the corresponding check box then select Actions->connect

EC2 Dashboard

Events

INSTANCES

Instances

Spot Requests

Reserved Instances

IMAGES

AMIs

Bundle Tasks

Launch Instance

Viewing: All Instance

+

Name	Type	State	Status Checks	Alarm Status
<input checked="" type="checkbox"/> SECC_R&D	t1.micro	running	2/2 checks passed	none

Actions

Instance Management

Connect

Get System Log

Get Windows Admin Password

Create Image (EBS AMI)

Add/Edit Tags

Change Security Group

Change Source/Dest. Check

Bundle Instance (instance store AMI)

Launch More Like This

Disassociate IP Address

Change Termination Protection

14. Enter the user name (ubuntu) and add the path of your .pem file and then click "Launch SSH Client".

Connect to an instance
Cancel

Instance: i-0465447a (SECC_R&D)

▶ Connect with a standalone SSH Client
▼ Connect from your browser using the Java SSH Client (Java Required)

Enter the required information in the fields below to connect to your instance. AWS automatically detects the key pair name, and public DNS for your instance. You need to enter location and name of the .pem file containing your private key.

Public DNS ec2-50-16-8-46.compute-1.amazonaws.com
User name: ubuntu
Key name: SECC
Private key path: C:\Users\lagamal\Desktop\SECC.pem
Example: C:\Users\username\Downloads\SECC.pem
Save key location: ☐ Stored in browser cache.

Launch SSH Client

Close

15. A new terminal will appear as if you are opening a terminal on a ubuntu machine.

```

ubuntu@domU-12-31-39-0E-32-64: ~ [105x27]
File Edit Settings Plugins Tunnels Help

System load: 0.0          Processes:           58
Usage of /:   9.6% of 7.87GB  Users logged in:    0
Memory usage: 6%          IP address for eth0: 10.192.49.142
Swap usage:   0%

Graph this data and manage this system at https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.

Get cloud support with Ubuntu Advantage Cloud Guest
http://www.ubuntu.com/business/services/cloud

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

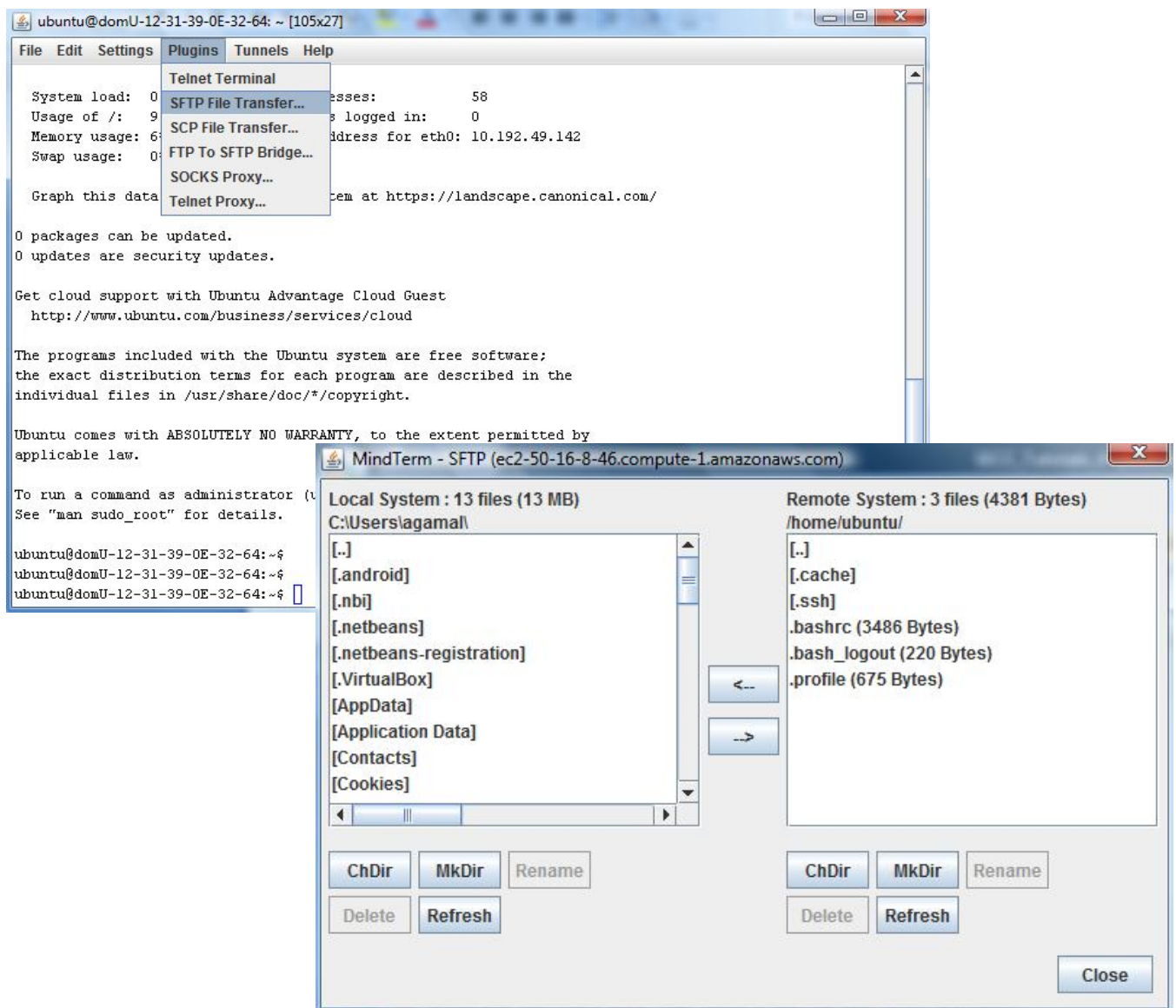
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@domU-12-31-39-0E-32-64:~$
ubuntu@domU-12-31-39-0E-32-64:~$
ubuntu@domU-12-31-39-0E-32-64:~$

```

16. You can now install whatever tools you want in the same way you are using a traditional Linux ubuntu machine. In addition to that if you want to transfer files, you can use the SFTP protocol as shown below:



By this way, we were able to use the Amazon EC2 IaaS technology to create a free ubuntu instance that acts as a server or computing unit for development and even commercial purposes. You can use it for hosting your website/web services or performing any processing required for your applications.

7. Summary

This tutorial briefly introduces the basic concepts of cloud computing. Moreover, it illustrates the common service models provided by the Cloud providers, and describes in details the famous Amazon EC2 web services focusing on the IaaS service model. A step-by-step explanation is given to demonstrate how Amazon EC2 can be used for developing applications without the need to pay for a complex infrastructure.

8. References

- [1] Reese, George (2009). Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: O'Reilly Media
- [2] <http://aws.amazon.com/documentation/gettingstarted/>
- [3] <http://www.appnexus.com/>
- [4] <http://www.rackspace.com/>
- [5] <http://www.gogrid.com/>

9. Abbreviations

AWS	Amazon Web Services
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol
BEEP	Blocks Extensible Exchange Protocol
SAN	Storage Area Network